Learning from visualizing and Interacting with the Semantic Web Dog Food

Christophe Gravier, Julien Subercaze

Learning from visualizing and Interacting with the Semantic Web Dog Food

Christophe Gravier and Julien Subercaze

LT2C, Télécom Saint-Étienne, Université Jean Monnet
10 rue Tréfilerie, F-4200 France
{christophe.gravier,julien.subercaze}
@univ-st-etienne.fr
http://portail.univ-st-etienne.fr

Abstract. Semantic Web conferences such as WWW and ISWC fostered a collaborative effort for the leveraging of Linked Data about conferences people, papers and talks. This effort gave birth to the Semantic Web Conference Corpus, a.k.a. the Semantic Web Dog Food Corpus. Many other conferences and journals contributed afterwards to this corpus, so that it is today a representative semantic data archive about our research community activities and progression. These metadata are consistent with Linked Data principles and therefore can be semantically processed by the machine. Although it is a matchless source of scientific knowledge for our community, it is difficult for the researcher, as a human, to browse this corpus that contains more than 180k unique triples. This paper presents our effort to bring a user-friendly Web application based on the Semantic Web Dog Food corpus that show the topics trends in Semantic Web research. The application was made freely available to the researcher as an end user. In this work we identify specific issues and barriers encountered when building the system, discuss how these were approached in this software, and how the lessons learnt can drive future implementations fostering the Web of Data.

Keywords: ontologies, application, corpus, architecture, metadata, conference.

1 Introduction

Over the last 10 years, tremendous research and engineering efforts were made in order to realize a Web that not only publishes unstructured data, but also structured and interlinked data ([12]). While the former Web was user-centric, the Web as for today is designed both for the human and the machine. In a decade, the Web turned into a data self-service, in which human and machine consumes the same raw ingredient (the datum), yet assembled using different recipes, made of data schemas for the machine, and presentation layers for the human. Over this journey, we have not only built one (possibly several) Semantic Web tower(s), but we have also gained knowledge on the laws that governs distributed data management, publishing, and consumption. Among several,

[13] stresses the discovery that factual knowledge is a graph, and terminological knowledge is a hierarchy, which is much smaller than the knowledge, and of low complexity. Moreover, while heterogeneity of data is unavoidable, we learnt that it is solvable. This scientific background, coupled with our engineering efforts to stack layers on top of URIs, is leveraging an increasing amount of semantic data (e.g. [3]). This trends seems to follow an exponential growth, given that the Web of things had gained interest in semantic sensing for smart environments [17].

Turning the Web into a distributed database (the so-called Web of Data) is achieved by connecting data using URIs and RDF. The Linked Data cloud diagram[1] is maintained as an effort for ontology reusability. Thanks to these tools, software agents have access to a large range of adapted Web services. Although these Web services are still struggling with multilingual data [8], the Semantic Web already lead to significant advances in real field applications [9], [11].

The Web of Data is therefore designed for the machine to process information in order to provide a service to the end-user, usually based on interlinking data from different sources, and/or thanks to the leveraging of implicit knowledge from linked data performed by a reasoner. As these datasets exponentially grow, the need for tools for human data consumers increases. As a consequence, information visualization and interactions are two key issues for Semantic Web data and Linked Data corpuses. In order to identify barriers that could be encountered when building such systems, we wanted to provide an information visualization and interactions tool over a recognized metadata corpus, which is the Semantic Web Dog Food corpus.
The goal of this paper is twofold: i) to provide a user interface for information visualization and interactions with the Semantic Web Dog Food, and ii) to discuss lessons learnt from building rich end-user Web applications that make use of Semantic Web and Linked Data.

This paper is organized as follows. Section 2 reports the genese and key figures about the Semantic Web Dog Food. Next, Section 3 presents the objectives and use cases covered by the targeted application. Section 4 focuses on the data visualization approach for our application, while Section 5 presents the interactions and the underlying architecture of the system. In Section 6, we discuss major barriers encountered and lessons learnt in building this application. Section 7 concludes.

2 The Semantic Web Dog Food

We are semantic researchers, but we are engineers also. In order to test the software developments that could be made using Semantic Web data, a common shared dataset as been proposed by [6]. This dataset is a corpus of triples that relates to past scientific conferences or workshops, along the authors and the organizations, which participated in these events. Recently, [16] proposes a Web

[1] `http://linkeddata.org/`

front-end for communities identification and as a data provider for social network of researchers. Since these are data from and to the Semantic Web scientists, it has been called the Semantic Web Dog Food corpus. For the purpose of this work, we have used the Semantic Web Dog Food corpus as its version of April 16th. It is the version that followed the integration of the metadata from the proceedings of the WWW 2012 international conference. The Semantic Web Dog Food corpus uses different Linked Data schemas. Among the core vocabularies of the corpus are **foaf**[2], **dc**[3], **geo**[4], and **ical**[5]. The Semantic Web for Research Communities vocabulary introduced by [4], which is prefixed by **swrc**.

An online illustration provides a more detailed view on how the vocabularies interact[6]. Our Semantic Web Dog Food instance is composed of 213,684 triples. Among 3,552 papers in the triplestore, 2,469 have a filled metadata for its abstract, and 2,187 papers are provided with a metadata value for a link to the PDF version of the paper.

It is also an interesting dataset as it presents some flaws which are similar to real field systems, most probably since its content is made of real-world semantic data. Unsurprisingly, hand-crafted semantic dataset are not perfect. While this may not have been the original will of the authors of the software, it is indeed part of the experiment to deal with incomplete or redundant data. In the next section, we expose the motivation to build such an application.

3 Motivations

We wanted to provide an interactive visualization Web application that bring the Semantic Web Dog Food to researchers as a user-friendly manner. We hope to gain a better insight of barriers encountered when building rich end-user applications based on the semantic Web. The Semantic Web Dog Food was an effort to bring semantic metadata processable by the Web of Data. This initiative was aimed at granting the machines the understanding of the contents that they are processing (people, venues, and articles). In the case of the Semantic Web Dog Food, this is only a corpus. The machine is able to crawl the provided RDFs dumps[7], and store it locally, or to perform SPARQL queries using the provided SPARQL endpoint[8]. We have yet to bring the end user interfaces that we can build upon such a corpus to the public, and to demonstrate how such interfaces can benefit to the end users, starting researchers as end users. Building a user-friendly Web interface was precisely the purpose of the WWW 2012 international conference metadata challenge [9]. We took up this challenge, and this paper

[2] `http://xmlns.com/foaf/0.1/`
[3] `http://purl.org/dc/elements/1.1/`
[4] `http://www.w3.org/2003/01/geo/wgs84\_pos\#`
[5] `http://www.w3.org/2002/12/cal/ical\#`
[6] `http://data.semanticweb.org/ns/swc/documentation/20071002-Properties.pdf`
[7] `http://data.semanticweb.org/dumps/`
[8] `http://data.semanticweb.org/sparql`
[9] `http://www.emse.fr/~zimmermann/metadata.html`

describes the system we have built. In order to list the application functionalities, we have encompassed two illustrative scenarios of this application.

3.1 Illustrative scenarios

Alice is a PhD student. Alice has started her PhD thesis a month ago. The research outcome of her work is still vague, yet she already has decided the main scope of her studies. She wants to contribute to close the gap between Big Data and the Semantic Web. As she was struggling for a month to find the relevant conferences covering her subjects of interest, Bob, her supervisor, pointed her to the Semantic Web Dog Food corpus. She is glad to find Linked Data on the corpus as it will allow her to have a single website that lists metadata and paper contents about articles that may be of interest. However, she would prefer to be able to browse the corpus per year, and to able to have some kind of representation of the evolution over the years of her topic of interest. She is however frustrated that she has to learn the SPARQL language, would she want to filter a subset of the Semantic Web Dog Food. It would also require her to know all the relevant keywords of the domains she study, yet she does not feel confident enough in knowing all relevant keywords for her work. Meanwhile, Bob asked her to write down few lines of comments on papers she read in a LaTeX file in order to help her building her bibliography page after page. However, given the Semantic Web Dog Food front end, she has to copy/paste all metadata of the relevant articles she read into her BibTeX file.

Bob is a researcher. Bob is a Professor. He is Alice's supervisor. He is widely recognized in the community for having proposed a new distributed reasoning algorithm. Recently, he had been asked to participate in a grant application. His contribution would be distributed reasoning over a stream of data. As he had apply his algorithm only on batch data so far, he has to dive into the literature on streams of data in order to encompass a proposal that adapts his distributed reasoning algorithm for stream of data. As he already knows about the Semantic Web Dog Food, he thinks of it as a good starting point. Unlike Alice, Bob is able to construct SPARQL queries to retrieve potentially interesting articles, yet he could prefer to avoid this in order to save time if possible. Moreover, where Alice as being inexpert needs to read the entire article to decide if it is relevant to her research of not, Bob is a confirmed researcher: from reading the abstract he can choose whether he will print of not the paper for further reading. Finally, Bob is targeting a very specific domain (stream of data), where Alice wanted to search for different and rather large topics such as Big Data and the Semantic Web in general. However, Bob also needs to build a bibliography out of its reading and to share it with its colleagues.

3.2 Functionalities

From Alice and Bob examples, the different functionalities that we want to provide to the end users are the following:

1. the system should be a user-friendly Web interface that provides a useful data visualization of the Semantic Web Dog Food corpus to Alice and Bob.
2. It must provide an unsupervised clustering algorithm to classify papers into research topics. This clustering algorithm must provide different configurations so that both users with coarse granularities clustering needs such as Alice, or fine grain needs such as Bob, could take the full advantage of the system.
3. It must take into account the temporal evolution of each research topics, per volume per year for instance, so that we can represent trends visualization.
4. the view of a selected topic should be twofold : at first with few information, so that experts such as Bob could fastly browse many papers in the selected topic, but also with the possibility to access further metadata so that both Bob and Alice could have a look of more details on a selected article (e.g. its abstract).
5. It must allow to build a bibliography of selected articles out of the Web front end so that Alice can make her report to Bob, and Bob share its state-of-the-art with his colleagues for when writing the grant application.
6. The system should serve several users seamlessly at runtime, which is a strong constraints given the computational time of SPARQL queries that may sometimes prevent near real-time querying when the data set is large.

From the previous list of functionalities that the system must provide, we have split them into two parts. The first three ones denote the visualization needs of the corpus for Alice and Bob, while the latter three are associated to the interaction needs with the corpus. In the next section, we will discuss the visualization needs, while Section 5 will expose how we provide interactions with the corpus for Alice and Bob.

4 Metadata visualization for the SWDF

4.1 The Semantic Web, data visualization, and streamgraphs

4.1.1 Semantic Web and Data Visualization Data provided by SPARQL endpoints are made for the machine to have a better description of Web contents so that they can provide a more adequate service to the end-users. Triplestores are underneath SPARQL endpoints. They are a collection of assertions over a given domain. Like most kind of corpus designed for the machine, Semantic Web corpuses are not designed to be processable by the human in their original form. Instead we need a translation form the formal language provided by RDF into natural language [20]. However, it is difficult for the human to read a corpus by reading as many sentences in natural language as there are assertions stored in a triplestore. That is the reason why the Semantic Web met Data Visualization [26]. Nevertheless, we explain below why we believe it is still only the very beginning of a long collaboration.

For a long time, data visualization over semantic data was supported by graph visualization. It is only natural to think to visualize semantic data using

graphs, for that is what they are [13]. This lead to famous representation of dense graph with thousands or millions of nodes, mainly also popularized as they are also used in Social Networks Analysis. There several available libraries that helps building such large RDF graph visualization [18], and examples of a resulting visualizations can be found online[10].

While it conveys some aesthetics, this figure illustrates that it fails to be a useful interactive visualization. It does help to gain an insight of the triples clusters in the graph, so that its density an be estimate roughly by the humans. However, how can Alice explore this graph easily and gain an insight at a higher level of what is inside the corpus? How can she filter relevant articles for a research problem? Using it, would Bob be able to observe the temporal evolution of topics he is interested in ? How Bob and Alice could browse at coarse grain between article while the visualization provide a fine grain visualization since each triple is encoded as a part of the observed graph ? Graph visualization actually fails to convey the browsing and coarse grain visualization concepts that the end users can expect, since the end-users have to perform a node-by-node walkthrough. This is the reason why we believe that data visualization over semantic corpus is an issue that goes beyond fitting a browsable graph into a screen [10]. While historically Data Visualization research has taken Social Network Analysis as a predominant application, we believe that Linked Data corpus will be an as interesting use case for Data Visualization, yet understudied today. However, there may be as many data visualization of a semantic corpus alternatives that they are end users needs. For the Semantic Web Dog Food corpus, we suggested a Streamgraph visualization to fit the visualization functionalities listed in Section 3.2 that we want to provide to Alice and Bob.

4.1.2 Streamgraph visualization Streamgraphs are considered as a kind of stacked graphs. It is designed to represent multi-layered data, along their evolution in time. Multi-layered data are data that are categorized, and whose volume changes with time. The data visualization offered by streamgraph helps in understanding both the classification of sets of data, as well as how the distribution of these sets over the entire corpus evolves. An example of streamgraph is given at Figure 1. Many more illustrations and explanations on streamgraph can be found at [19]. The first proposal that was close to a streamgraphs is certainly the *Theme River* system [1]. *Theme River* is a "prototype system that visualizes thematic variations over time across a collection of documents". In the case of *Theme River*, the document were documents from Fidel Castro speeches, articles, etc., over a 40-years period. Theme River is based on the metaphor that a river flow conveys the concept of the passage of time. In the provided streamgraph example, the curves provide the aesthetics of a river flow. The Streamgraph term was coined in 2007 by [14], who made an attempt of a new stacked graph visualization applied to Last.fm, a music listening online service.

[10] `http://www.mkbergman.com/wp-content/themes/ai3/images/2008Posts/080128_mkbergmanweb.png`

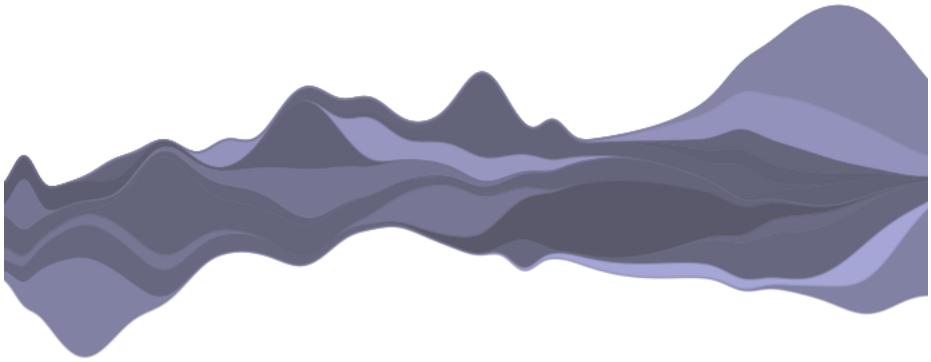

Fig. 1. Example of a streamgraph.

It was next popularized by the protoviz [11] javascript library, and brought mainstream when in February 2008, the New York Times published an unusual chart of box office revenues for 7500 movies over 21 years. While the figure may not be freely reproducible in this paper, a free interactive version is offered by the New York Times online[12].

In a streamgraph, each layer represents an object of study, and the stacked view of layers provides both their relative distribution and their absolute evolution in time. The relative distribution is provided for a given date given the height of each layer relatively to the others. The absolute evolution in time is provided by the relative height of the stacked layers between each point in time in axis.

For the illustrative use cases of Alice and Bob, streamgraphs can be a great semantic web visualization over the Semantic Web Dog Food corpus. Using different streamgraphs for different granularity of papers categorization, we could provide to Alice and Bob a synoptic view of topic of their interests in the corpus, along their evolution in time. This however implies that we are able to classify each paper in the Semantic Web Dog Food as different topics, which we do not know in advance, due to the absence of a topic ontology about semantic conference papers. In Section 4.2, we discuss the approach built on the Latent Dirichlet Allocation algorithm that we took in order to categorize Semantic Web Dog Food papers.

4.2 Latent Direchlet Algorirthm (LDA)

The Latent Dirichlet Allocation [5] (henceforth LDA) is an algorithm that is widely recognized for identifying different topics in a set of documents. The

[11] `http://mbostock.github.com/protovis/`

[12] `http://www.nytimes.com/interactive/2008/02/23/movies/20080223_REVENUE_GRAPHIC.html`

LDA approach assumes that an entire corpus of documents is composed of a fix number of topics, which is an input of the model. That means that given a fix number of topics noted k given as an input to the algorithm, the LDA algorithm tries to find k topics that are prominent in the corpus. It assumes that a topic is characterized by a distribution over words, and that a document is modeled as a bag of words (word ordering does not affect the result). Since a document is a mixture of topics in this model, it cannot be classified as a clustering algorithm: since a document can be associated with multiple topics, yet under different probabilities. For the same reasons, a word can belong to several topics. The performance of the LDA algorihtm has made it a relevant choice for unsupervised topic learning problems. Although it has originally be designed to work over an entire corpus, online learning adaptation of the LDA algorithm has been proposed [2].

4.3 LDA, streamgraph, and the SWDF

The LDA algorithm can be used to infer top k topics from a corpus, where k is an input parameter. The cornerstone of our visualization system is to use the LDA algorithm on the Semantic Web Dog Food corpus in order to identify the major topics in our research community, and then to build the statistics out of these topics in order to build an *ad hoc* streamgraph. Each layer of the streamgraph would therefore represent one of the k prominent topics in the SWDF corpus. However, the LDA algorithm takes a set of text documents as in input, and not Semantic Web data. We could use the text of each article to build our text corpus. Nonetheless, not all the articles present in the SWDF provide a Linked Data for the document content itself. Moreover, the document can also be noisy speaking of the different topics it covers. At the opposite, we believe that the abstract of the documents conveys the major topics covered by the article, as well as few superfluous texts. Meanwhile, there are more Linked Data about abstracts of article than linked data for the entire document.

As explained in Section 4, we will use the input parameter k of the LDA algorithm to provide different granularities of visualization. In this section, we illustrate this process by fixing k to the value 10. Under this setting, we are therefore trying to build a streamgraph of 10 layers. Running the LDA algorithm with $k = 10$ gives 10 topics as provided in Tab 1. Word ordering is significant in this table, as each topic corresponds to a distribution of words for this topic. The more a word occurs early in the sequence of word for a topic, the more it has been identify as significant for this topic. We believe that the results are very acceptable, as the output present 10 different and relevant topics out of the articles published by our community over the last 5 years. In order to build the streamgraph, we affect each paper to the topic it mostly belongs to. Once this classification is done using the result of the LDA algorithm, we can build the statistic of the number of papers per year per topic. At this point, we inferred all the necessary information to build the streamgraph out of the abstracts of the articles present in the SWDF.

#	Top 10 words in the distribution	papers
0	search, web, information, query, results, queries, user, users, paper, based	316
1	ontology, ontologies, knowledge, semantic, paper, reasoning, approach, owl, concepts, domain	338
2	speech, corpus, system, recognition, language, paper, video, spoken, database, dialogue	112
3	language, corpus, paper, annotation, text, resources, lexical, evaluation, corpora	367
4	data, rdf, query, queries, sparql, web, graph, approach, graphs, processing	183
5	web, mobile, content, applications, pages, sites, users, page, browser, user	89
6	data, web, semantic, information, research, paper, metadata, knowledge, system, applications	512
7	problem, data, algorithm, method, show, learning, model, based, set, paper	210
8	service, web, semantic, paper, model, applications, approach, application, business, events	166
9	social, network, users, information, content, online, news, user, study, twitter	176

Table 1. Word and volume of papers distributions over ten topics (2,469 papers).

It is important to note that the LDA algorithm is not deterministic [5]. Under the same parameter, it cannot guarantee to produce the same words distribution for each topic, even the same bag of words for each topic. Moreover, as the topic ordering in the output is not significant for the LDA algorithm, it leads to different streamgraphs would the topic ordering differs. However, over multiples runs, the LDA algorithm gave very similar results. We grant this results to the high number of iterations (1,000) we set for each algorithm run.

5 Interacting with the SWDF

Streamgraph are originally a none-interactive visualization. As they became popularized by javascript library such as protoviz and its fork d3 [13], they were granted interactive functionalities. This interaction is mainly supported by rendering each layer of the streamgraph as clickable, in order to update part of the DOM tree of the webpage. In our visualization over the SWDF, each layer represents a topic. Interactivity is required in order to update the Web page so that when selecting a topic from its layer in the streamgraph, the list of the articles classified in this topic is displayed. Modern approaches for interactivity in the Web browser make heavy use of the javascript programming language. However, the Linked Data attached to a paper are stored server-side in a triplestore. This is the case if we would have been using directly the SWDF SPARQL endpoint, or using our duplicate of the SWDF corpus. In order to provide this function-

[13] `http://d3js.org/`

ality, we build an entire architecture, from the triplestore to the streamgraph interactions with the end user.

5.1 Architecture

We want to provide to Alice and Bob with different granularity speaking of number of topics, noted k, as delivered by the LDA algorithm. Because the LDA algorithm is not deterministic and require some time to compute its output, we had to set in advance different values for k. We have chosen three different configurations for k :

- $k = 10$ for a coarse-grained visualization since the number of topics for the thousands articles present in the SWDF is limited to 10. Alice should have interest in this configuration would she had to firstly understand the subjects covered by Semantic Web research, and how they are distributed among the years.
- $k = 20$ a mid-size value for k, since it already provide more specific topics, whereas the streamgraph is still very usable.
- $k = 30$ for a fine-grained visualization. We thought Bob could have interest in this visualization if he wants to build a state-of-the-art on a very specific topic.

The system is twofold:

- A first script sets up the trends clusters for each configuration (10, 20, and 30) by extracting metadata from a local instance of the Semantic Web dog food using Empire software[14] (a.k.a. "JPA+RDF"). This script afterwards publishes statistics for each configuration a JSON file, so that the browser could download them when rendering the Web Page in order to build the streamgraph using the d3 javascript library[15]. Another JSON file[16] stores all statistics for server-side usage. This step is performed only once for each update of the SWDF corpus.
- A second software deals with runtime interactions, and serves the Web application to Web browsers using Apache Tomcat[17]. The browser performs AJAJ (Asynchronous Javascript And Json) queries using jQuery[18] in order to update the client browser. Server-side, queries are processed by a RESTful API [15] based on the Java JSR 311 [19] using the Jersey[20] implementation.The runtime architecture of the system is depicted at Figure 2.

[14] `https://github.com/mhgrove/Empire`
[15] This file can be browsed online at `http://www12-satin.telecom-st-etienne.fr/output-dist.json`
[16] `http://www12-satin.telecom-st-etienne.fr/output-srv.json`
[17] `http://tomcat.apache.org/`
[18] `http://jquery.com/`
[19] `http://jcp.org/aboutJava/communityprocess/final/jsr311/index.html`
[20] `http://jersey.java.net/`

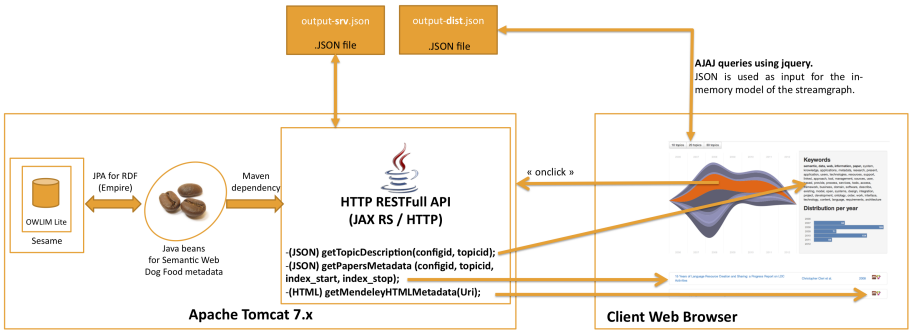

Fig. 2. Runtime architecture for the visualisation and the interaction with the SWDF.

5.2 Additional features

Linked Data allowed us to provided additional functionalities. Through three distinct icons for each listed paper, we provide the following additional features to our system :

- A link to the corresponding SWDF entry (we are end users, but also researchers)
- A link to add the entry to the end user's Mendeley[21] account. Mendeley is a free online bibliography service.
- A link to the external PDF file of the article, if available in the SWDF. If not, a gray tint is applied to the icon.

The application is accessible at `http://www12-satin.telecom-st-etienne.fr`. A capture is provided at Figure 3. At the right of the streamgraph, a portlet presents the topic that the user selected, that means the sequences of the first 50 words that best depicted this topic. The threshold of 50 words was chosen for user experience purpose The portlet also present the temporal evolution of the selected topic, speaking of number of paper per year classified in this topic.. Below the streamgraph and the topic description, a list of the first 10 papers, alphabetically sorted, belonging to this topic appears once the user select a topic by clicking a layer in the streamgraph. All other papers are browsable through pagination provided at the end of the page.

6 Discussion and lessons learnt

From this software development, we are not only providing a visualization and interactive application over the Semantic Web Dog Food, but also we have gained a better insight on barriers for developing such Semantic Web front ends. In this section, we are discussing each of the main issue we have encountered.

[21] `http://www.mendeley.com`

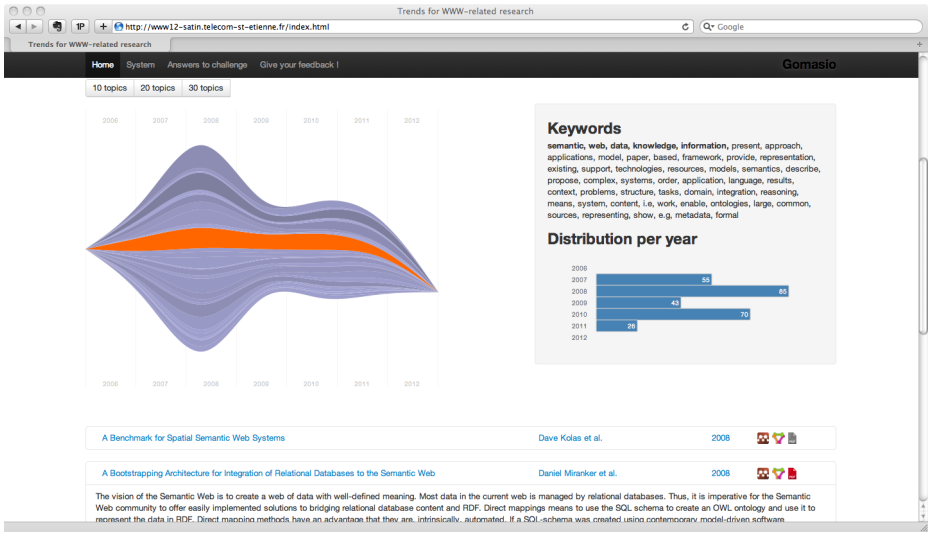

Fig. 3. Visualisation and the interaction with the SWDF Web application.

6.1 Lesson #1: The Semantic Web Dog Food gone bad

The Semantic Web Dog Food has been created in order to dispose of a shared real word corpus for illustration and evaluation purpose. There is no denying that the corpus have fulfill this objective and even beyond. This was lately illustrated at the WWW 2012 metadata challenge[22].

Meanwhile, Semantic Web applications are nowadays distributed. This is why the Linked Data, along the Read-Write Web (c.f. Section 6.3), are critical issues for the Web. Many other issues arise from a distributed architecture for the Semantic Web, such as interoperability (partially supported by the Linked Data cloud), scalability, parsing [22], discovering services, pingback architectures, discovering **owl:sameas** relationships, provenance .... Yet our own dog food is still a unique centralized server, whose updates are performed by the human, from Excel exports. It no longer represents the Web of Data in its current evolution. We think that the dog food for our community has to evolve into a distributed architecture, where each university exposes its own Semantic Data from its publications. This would better matches the Web of Data approach, as well as providing a common ground for testing, and evaluation of the current research in our community.

We therefore urge on Semantic Web researchers to provide their own running instance of a dog food server. Would several research centers publish their own data about people, conference, and publications, we would be closer to the reality of the Web of Data. Obviously, it is difficult for each university to fund the

[22] `http://www.emse.fr/~zimmermann/metadata.html`

software development of its own dog food. In order to foster a "distributed semantic web dog food", we are providing a recipe for cooking your own dog food[23]. The recipe guide you step-by-step to install and configure an OWLIM triplestore and the Sesame framework, along the **bibtex2rdf** application that we have built[24]. As we, as researchers, are generally used to maintain the bibtex files of our publications, the **bibtex2rdf** software takes as input the list of bibtex files on the Web or on the filesystem of all the people in the research team, lab or univeristy, and, at a given interval of time as a *crontab* parameter, updates the triplestore with new publications, if any (i.e. if the bibtex file were updated). We are trying to maintain a list of existing distributed dog food nodes[25]. By providing the tool to create a distributed dog food, we hope to foster the creation of a playground for testing and demonstration of modern issues in the Web of Data.

6.2 Lesson #2: A little bit of cache goes a long way

The system proposed at Section 5.1 is deployed as a virtual machine, which is provided with 2 Gb RAM and 2 CPUs. It suffers at runtime from a high response time where it comes to performed pagination since all papers in a topic are not displayed in the same Web page. Given a page number corresponding to the selected page by the end-user, the system translates this number into a range of papers, and extracts the URI for this range for the selected topic. For each URI, metadata associated to the article are serialized in JSON, so that the browser can display them for the current page. Server-side, retrieving the metadata for each URI, means to perform a SPARQL query for this URI. The SPARQL query is generated by the Empire[26] framework, which is a Java Persistence API for RDF, that use SPARQL for providing persistency. Generating several SPARQL queries with a filter in a row is not very efficient when running on a commodity server. While using SPARQL for persistence guarantees interoperability of the persistence layer, it may not be the most efficient one. There is also no denying that the evolution of the SPARQL standard is raising some issue as it is becoming more and more complex [25]. Relying on semantic data indexation like provided in [22] may be a hint to improve the general performance of persistence services for semantic data.

As for now, just like a little Semantics goes a long way [21], caching is the Semantics companion on its journey over the Linked Data Cloud. Caching is very interesting in Semantic Web applications due to the property of dereference ability of URI, along the monotonic property of RDF/OWL knowledge bases. Dereferencability of URI makes it possible to elaborate a URL-based caching policy, which is easy to setup and deploy. The monotonic property of RDF/OWL

[23] link to recipe
[24] The source code of ths application can be downloaded at : `https://github.com/cgravier/bibtex2rdf`
[25] `http:/wherewestorethislist`
[26] `https://github.com/mhgrove/Empire`

guarantees that it is an interesting option since the semantic data in cache is less subject to changes, which means the more efficient the cache. Such a caching policy is also very convenient to setup and deploy in most common reverse proxy servers such as Apache Web server, or nginx[27]. Using caching mechanism (Apache Web server configured as a reverse proxy on our RESTFul API that listen to HTTP GET queries), our application is 5 and up to 10 times faster than the version without caching. A little bit of cache goes a long way.

6.3 Lesson #3: We must decrease the amount of boilerplate code for the Web of Data to scale at development time

In the Web of data, semantic data are to be retrieved over HTTP. Localization of the resource is provided thanks to the dereference ability of the URI of the data. This calls for RESTFul Web Services, like the ones deployed in our system (c.f. 5.1). Traditionally, RESTFul APIs are the cornerstone for a loosely coupled architecture. It however has to be implemented for each vocabulary of the data, would it be local vocabularies or vocabularies from the Linked Data cloud. *Ad hoc* Restful Web Services over Linked Data are an hindrance for the Web of Data. Since all these Restful Web Services over a triplestore have in common to fetch or push data based on their data, an emerging paradigm is to use HTTP directives[28] as performatives for updating and querying the triplestore.

The first step towards this goal was made by the end of 2011 with the creation of the Read Write Web software[29], that implements a standalone HTTP server over netty[30]. Listening for HTTP requests, the server serializes and desterilizes data into the filesystem using turtle (or n3) notation. Recently, the Read Write Web software was enhanced with a WebID module[24], which makes it very compatible with modern Semantic Web architecture. We have successfully used it in a distributed Semantic Social Network use case[23].

It is however not yet a standard. Nonetheless, and in the same time of the Read Write Web software development, this emerging paradigm has lead to the submission to the W3C of the Linked Data Basic Profile 1.0[31], on March 26th 2012 by IBM. This submission provides the effort for the standardization of the paradigm. We believe that this paradigm not only encompass read and write RDF data over HTTP, but could also be expanded for providing autonomic creation and deployment of Semantic Web services in the Web of Data. Just like persistence frameworks in Java or PHP provide the automatic creation of methods like finders, this could be achieve but at the Web of Data scale. Software like the Read-Write Web could implement the generation of common complex services over HTTP, such finding all instances of a **rdf:class**, find all instances by key, etc.

[27] `http://wiki.nginx.org/`
[28] `GET,POST,PUT,DELETE,HEAD,PATCH`
[29] `http://dvcs.w3.org/hg/read-write-web/`
[30] `www.jboss.org/netty`
[31] `http://www.w3.org/Submission/2012/SUBM-ldbp-20120326/`

In 2012, we are in the process to provide standardization and implementation of tools for the developer to leverage Semantic Web services in the most efficient way, just like we spent the last years to leverage user-generated contents. Boilerplate code for writing Semantic Web applications must decrease.

7 Conclusion

In this paper, we provide a description of a system for visualizing and interacting with the Semantic Web Dog Food corpus. It has been designed with the will to provide a useful application for the researcher, as an end-user. The visualization relies on a streamgraph that presents layers of topics discovered by running the Latent Dirichlet Allocation algorithm over the abstracts in the Semantic Web Dog Food corpus.

From this experience, we stressed several issues. First, as for the Semantic Web Dog Food itself, we advocate an evolution of the corpus towards a decentralized architecture. This would allow to setup a fragment of the Web of Data for the researchers to experiment and evaluate on it. Moreover, from a more technological perspective, we pointed out the need of caching when it comes to build a RESTFul API on top of a triplestore. Finally, we discuss leveraging a generic read/write Web of Data approach, agnostic to RDF vocabularies, yet that could be the processor of HTTP queries for fetching, deleting and inserting triples. We presented two software contributions in this article. The first one is the visualization and interactive Web application for researchers as end-users for the Semantic Web Dog Food. The second is a recipe for setting up your own distributed Dog Food server. We can only encourage you to cook your own dog food. In future works we hope to contribute to the Read Write Web by implementing advanced features such as auto generation of finders or named queries, in order to decrease the boilerplate code required to realize a node in the Web of Data.

Acknowledgments. This work has been funded by Conseil Général de la Loire (CG42).

References

1. Havre, S., Hetzler, Whitney, P., B., Nowell, L.: ThemeRiver: Visualizing Theme Changes over Time. ThemeRiver: visualizing thematic changes in large document, vol. 8, issue 1,pp. 9-20, IEEE (2002).
2. Canini, K.R., Shi, L., Griffiths, T.L.: Online Inference of Topics with Latent Dirichlet Allocation. Journal of Machine Learning Research - Proceedings Track, pp. 65-72 (2009)
3. Corlosquet, S., Delbru, R., Clark, T., Polleres, A., Stefan, D.: Produce and Consume Linked Data with Drupal! In: Proceedings of the 8th International Semantic Web Conference, LNCS, vol. 5823, pp.763–778, Chantilly, VA, Springer, Heidelberg (2009).

4. Sure, Y., Bloehdorn, S., Haase, P, Hartmann, J., Oberle, D.: The SWRC Ontology - Semantic Web for Research Communities. In Proceedings of the 12th Portuguese Conference on Artificial Intelligence - Progress in Artificial Intelligence (EPIA 2005), LNCS, 3803, pp. 218–231, Covilha, Portugal. Springer, Heidelberg (2005).
5. Blei, D. M., Ng, A. Y., Jordan, M. I.: Latent Dirichlet Allocation. Journal of Machine Learning Research. 3, 993–1022 (2003).
6. Knud, M., Heath, T., Siegfried, H., John, D.: Recipes for Semantic Web dog food - The ESWC and ISWC metadata projects. In: Proceedings of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference (ISWC2007+ASWC2007), 11-15 Nov, Busan, South Korea. Springer, Heidelberg (2007).
7. Hogan, A., Harth, A., Umbrich, J., Kinsella, S., Polleres, A., Decker, S.: Searching and browsing Linked Data with SWSE: The Semantic Web Search Engine. Web Semantics: Science, Services and Agents on the World Wide Web, Volume 9, Issue 4, pp. 365–401, ISSN 1570-8268, 10.1016/j.websem.2011.06.004 (2011).
8. Gracia, J., Montiel-Ponsoda, E., Cimiano, P., Gmez-Prez, A., Buitelaar, P., McCrae, J.: Challenges for the multilingual Web of Data, Web Semantics: Science, Services and Agents on the World Wide Web, Volume 11, pp. 63–71, ISSN 1570-8268, 10.1016/j.websem.2011.09.001 (2012).
9. Hansch, D.: Practical applications of Semantic Wikis in commercial environments. In: Proceedings of the 10th International Semantic Web Conference, 23-27 oct. Bonn, Germany, LNCS, vol. 7032, Springer, Heidelberg (2011).
10. Howse, J., Stapleton, G., Taylor, K., Chapman, P.: Visualizing ontologies: a case study. In: Proceedings of the 10th International Semantic Web Conference, 257–272 oct. Bonn, Germany, LNCS, vol. 7031, Springer, Heidelberg (2011).
11. Mcginnis, J.: Why Semantics Makes Sense for News Agencies. In: Proceedings of the 10th International Semantic Web Conference, 23-27 oct. Bonn, Germany, LNCS, vol. 7032, Springer, Heidelberg (2011).
12. Auer, S., Lehman, J.: Creating Knowledge out of Interlinked Data. Semantic Web, IOS Press, 1(1), pp. 97–104 (2010).
13. van Harmelen, F.: 10 years of semantic web research: searching for universal patterns. In: Proceedings of the 10th International Semantic Web Conference, 23-27 oct. Bonn, Germany, LNCS, vol. 7032, pp. 389, Springer, Heidelberg (2011).
14. Byron, L., Wattenberg, M.: Stacked Graphs Geometry & Aesthetics. Transactions on Visualization and Computer Graphics, vol. 14, no. 6, pp. 1245–1252, IEEE (2008)
15. Fielding, R. T., Taylor, R. N.. Principled design of the modern Web architecture. ACM Transactions on Internet Technology (TOIT), vol. 2 Issue 2, (2002)
16. Monaghan, F., Bordea, G., Samp, K., Buitelaar, P.: Exploring Your Research: Sprinkling some Saffron on Semantic Web Dog Food. Semantic Web Challenge at the International Semantic Web Conference, Shanghai, China (2010).
17. Hasan, S., Curry, E., Banduk, M., O'Riain, S.: Toward Situation Awareness for the Semantic Sensor Web: Complex Event Processing with Dynamic Linked Data Enrichment, In: 4th International Workshop on Semantic Sensor Networks, pp. 69-81, Bonn, Germany (2011).
18. Large-scale RDF Graph Visualization Tools, `http://www.mkbergman.com/414/large-scale-rdf-graph-visualization-tools/`
19. Making sense of streamgraphs, `http://www.visualisingdata.com/index.php/2010/08/making-sense-of-streamgraphs/`
20. Kaljurand, K; Fuchs, N E (2007). Verbalizing OWL in Attempto controlled English. In: Proceedings of OWL: Experiences and Directions (OWLED 2007), Innsbruck, Austria (2007).

21. Hendler, J.: The dark side of the semantic web. IEEE Intelligent Systems 22(1), 24 (2007)
22. Fernàndez, J. D.: Binary RDF for Scalable Publishing, Exchanging and Consumption in the Web of Data. In: Proceedings of the 21st International World Wide Web Conference, WWW 2012, Lyon, France- In Press. (2012).
23. Story, H., Blin, R., Subercaze, J., Gravier, C., Maret, P.: Turning a Web 2.0 Social Network into a Web 3.0, distributed, and secured Social Web application. In: Proceedings of the 21st International World Wide Web Conference, WWW 2012, Lyon, France - In Press. (2012).
24. Story, H., Harbulot, B., Jacobi, I., Jones, M.: Foaf+ssl: Restful authentication for the social web. In: Proceedings of the First Workshop on Trust and Privacy on the Social and Semantic Web, co-located with ESWC2009, the 6th European Semantic Web Conference, June 1st, 2009 - Heraklion, Greece (2009)
25. Arenas, M., Conca, S., Pérez, J.: Counting Beyond a Yottabyte, or how SPARQL 1.1 Property Paths will Prevent Adoption of the Standard. Proceedings of the 21st International World Wide Web Conference, WWW 2012, Lyon, France - In Press. (2012).
26. Oren, E., Delbru, R., Decker, S.: Extending faceted navigation to RDF data. In: Proceedings of the 5th International Semantic Web Conference, 5-9 nov., LNCS 4273, pp. 559-572, Athens, GA, USA (2006).